

Case study

Self-service ticket upgrades

Fever

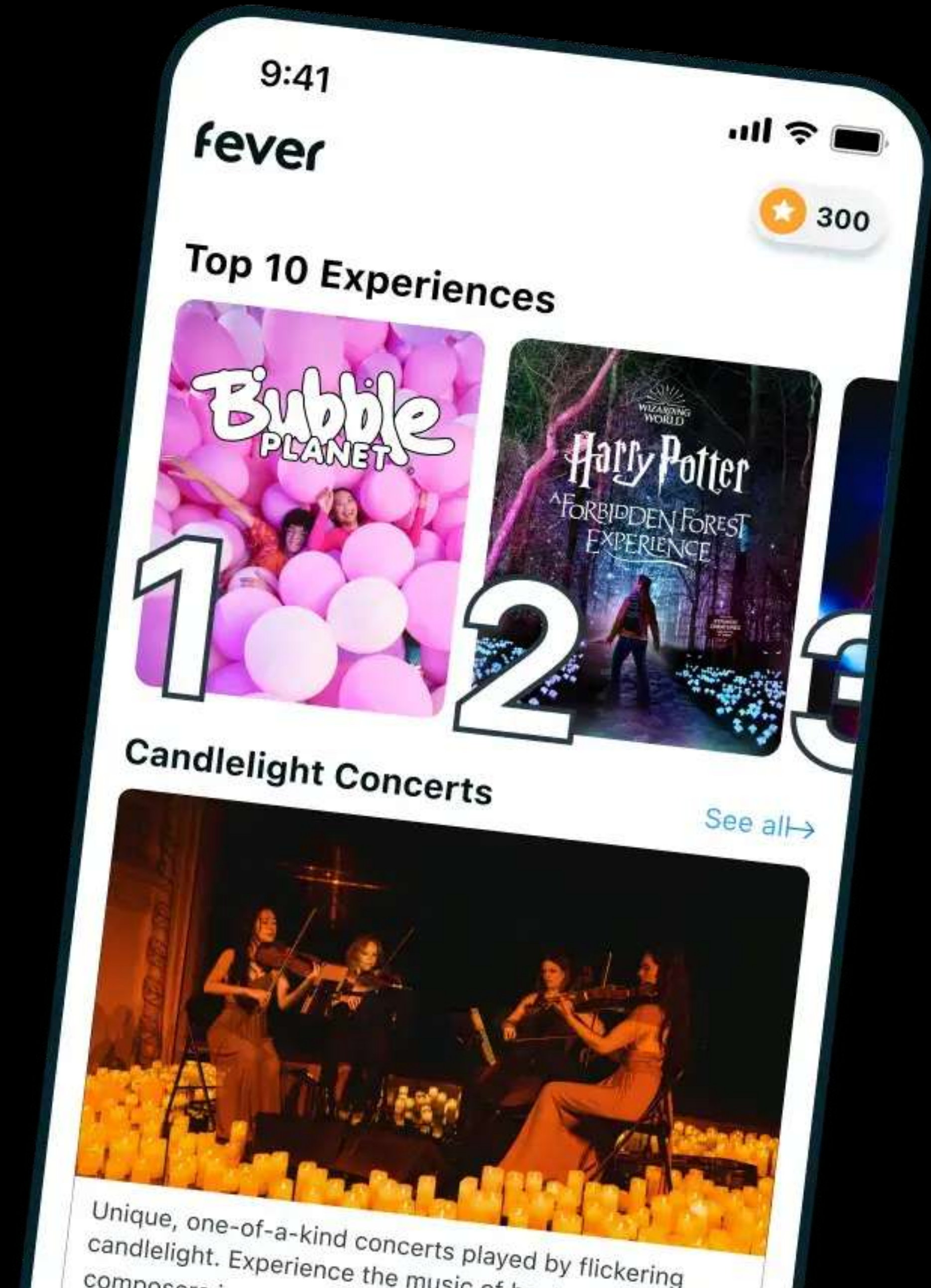
Role
Senior Product Designer

Year
2026



Context

Fever aims to let users discover, plan, and book local events, offering personalized recommendations based on their location and interests across activities like music, exhibitions, food, and nightlife.



Context

Fans who wanted to move from General Admission to VIP had to go through User Operations.

That meant more support tickets, manual coupon handling, and drop-off while people waited for a representative.

Upgrade tickets



Make your experience even better!



Transfer tickets



Going with a friend? Send them their ticket



Change date and/or time

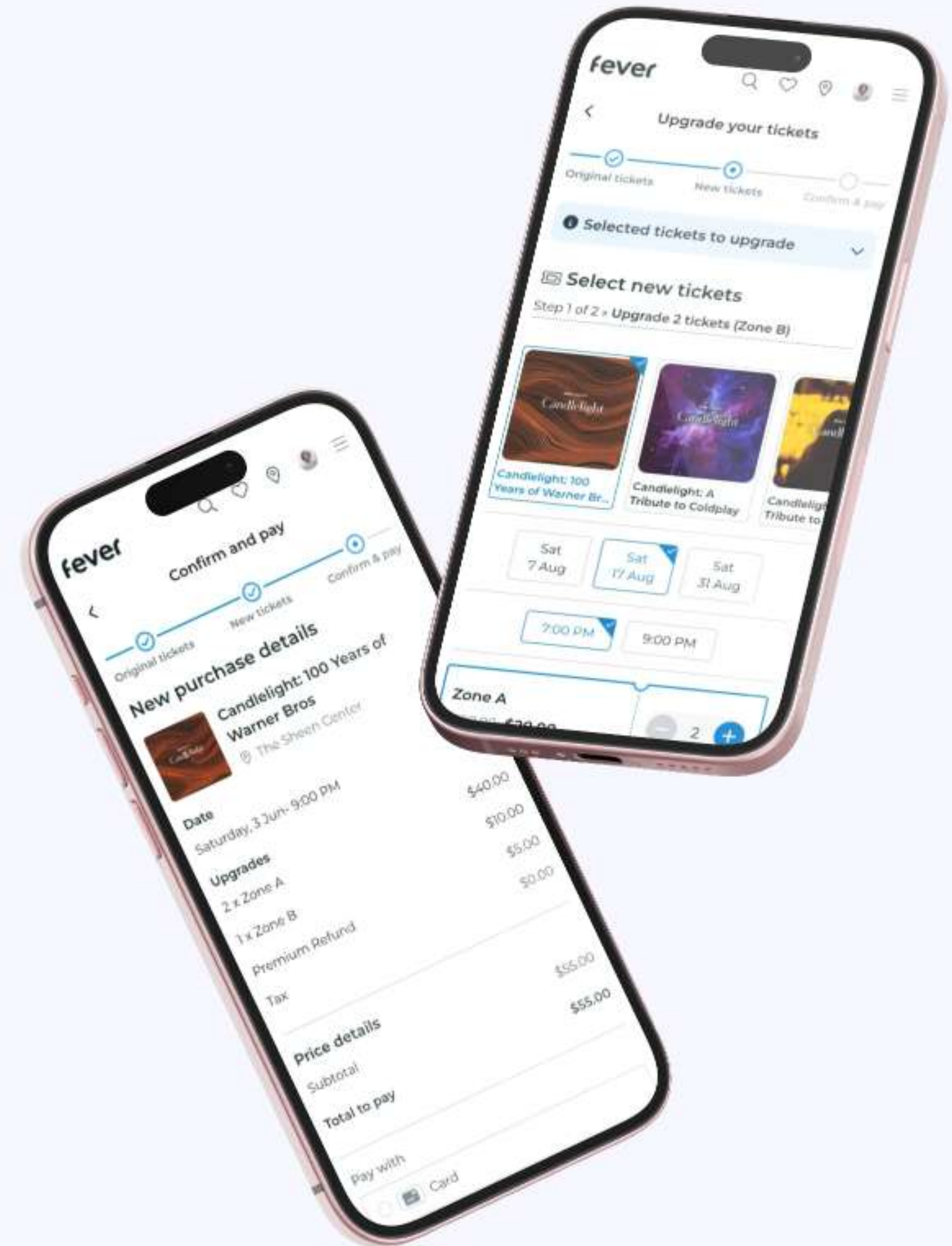


Cancel your current tickets and select a new date/time



Challenge

Transitioning Fever from manual, support-heavy ticket exchanges to a frictionless, product-led upgrade path.



Problem Space

The initial phase focused on gaining a clear understanding of business expectations and user needs.

The User Operations Audit

To define the MVP's requirements, I led an audit of our **User Operations logs** to understand why fans were dropping off during the exchange process.

- A significant volume of support tickets originated from fans wanting to **move from General Admission to VIP tiers**.
- **High friction** was identified in **manual exchanges** involving support tickets and manual coupon handling, which caused high drop-off as users waited for a representative.
- User feedback frequently cited "**confusion over total costs**" as primary blocker when dealing with manual agents.
- Needs to **reduce manual coupon handling** and support tickets for User Operations (Stakeholder)

*"I just arrived and the venue looks amazing.
I want a better view"*

*"I bought GA to save money, but now I've
changed my mind and want the VIP experience"*



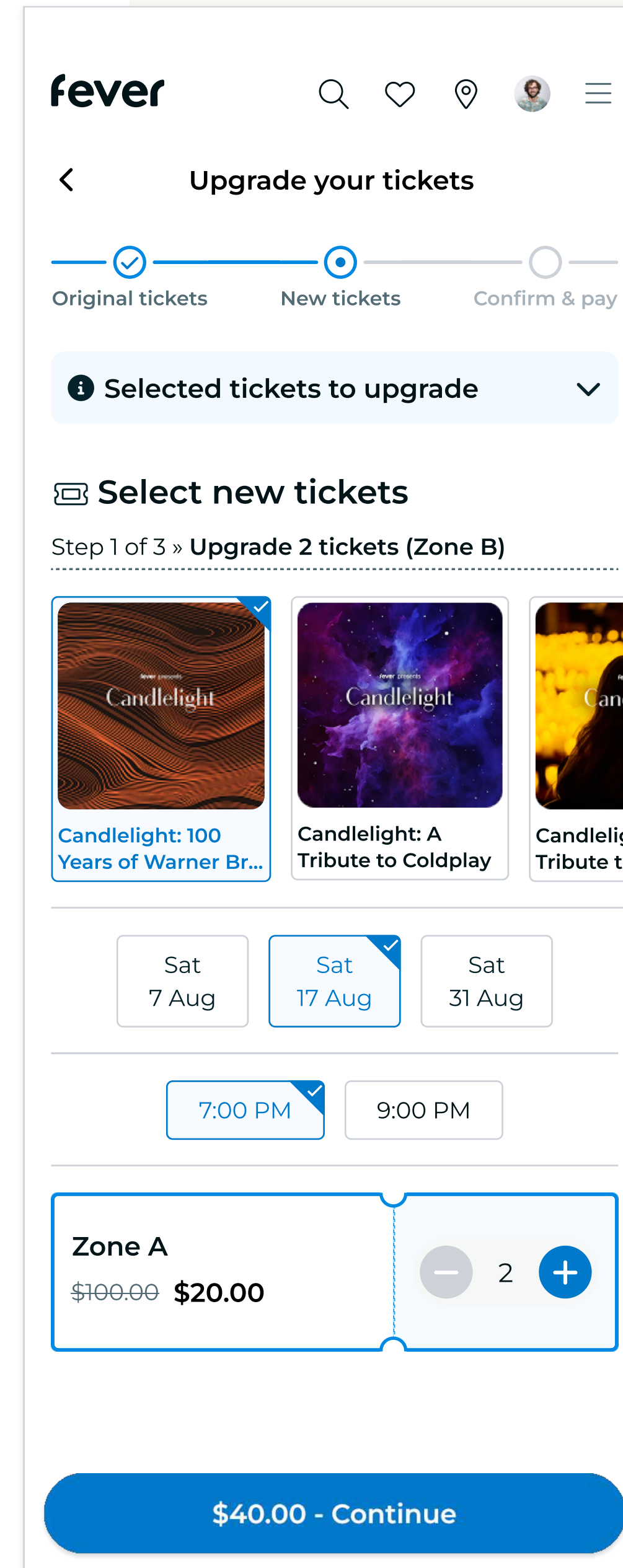
Real-Time Requirement Definition Through AI

I integrated AI (specifically LLMs and Cursor) across four distinct phases of the Fever project to bypass traditional bottlenecks and ensure technical feasibility from day one.

Instead of waiting for the PM to define the PRD and requirements upfront, we **leveraged AI to analyze and summarize the collected data**, allowing us to **quickly build a live prototype** and collaboratively shape the PRD.

This approach enabled **real-time triage communication** between the PM and the Dev Lead, helping us **define requirements and solution characteristics** iteratively, as we progressed.

Smart Restrictions: We identified "Smart Restrictions" (e.g., upwards-only pricing) much earlier in the process, preventing late-stage redesigns.



Key features

Requirement Category	Requirement	User Story	Priority	Status
Configurable Settings	Feature Enablement	As Fever staff, I am able to enable and disable upgrades at plan level	High	In progre...
Configurable Settings	Same Day Restriction	As Fever Staff, I can enable a limit for the upgrades feature to the same day as the original order	Medium	In progre...
Configurable Settings	Same Time Slot Restriction	As a Fever Staff, I can limit upgrades to new tickets only within the same time slot as original ones	Medium	In progre...
Configurable Settings	Higher Price Restriction	As Fever Staff, I am able to limit the upgrade of tickets to higher ticket price of the same plan ID	High	In progre...
Configurable Settings	Liv Golf Exclusive Requirement (where ticket types within time slot correspond to days)	As Fever Staff, I can block the upgrade when several ticket types in the same time slot to a pack	High	In progre...
Configurable Settings	Cross Plan Upsell	As Fever Staff, I can enable upgrades to several plan IDs	High	In progre...
Configurable Settings	Session Type Enable	As Fever Staff, I am able to disable the feature at session type level	Very L...	Out of Sc...
Configurable Settings	Inclusion of upgrades in Plan Creation Process	As Fever Staff, I can enable and disable upgrades in the	Medium	Launched

Balancing User Needs with System Stability

To ensure a safe rollout, we initially **limited the feature to plans** with fewer than 20 sessions per slot and no complex seating maps (e.g., Candlelight concerts).

Phase 0, Internal Testing on 3–5 plans (Candlelight / SSI).	INTERNAL
Phase 1, Pilot 20–30 Candlelight plans in a single city.	PILOT
Phase 2, GA Global availability across all eligible Marketplace plans, once the pilot clears the bar.	GA

Upgrades and rescheduling for plans with real seating maps, and finer-grained PRP so people can split or tweak protection per line during an upgrade.



Three core pillars

- **Clear Entry Points:** Discovery is streamlined through the “My Tickets” area and post-purchase CRM emails. Users gain full visibility into which tickets are eligible for upgrade through a dedicated, intuitive interface.
- **Tier Selection:** A clear and guided choice architecture enables users to easily select which specific tickets they want to upgrade, reducing friction and decision complexity.
- **Delta Payment:** Users pay only the price difference when upgrading, eliminating the need for refunds and repurchasing, and ensuring a smoother, more efficient transaction experience.

Upgrade tickets
⚡ Make your experience even better! >

Upgrade 1 ticket (Zone C)

Zone A
~~\$100.00~~ \$50.00

− 0 +

Zone B
~~\$40.00~~ \$10.00

− 1 +

\$10.00 - Continue

Solution

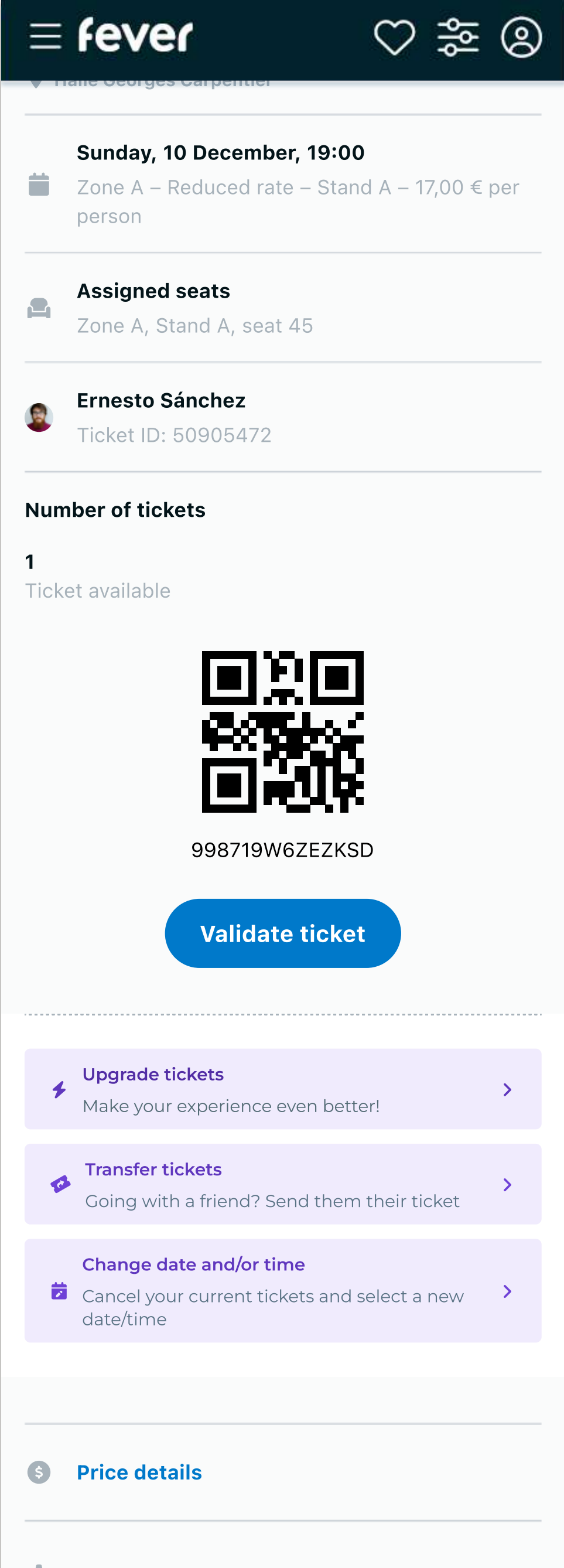
Into a new and sped up process through direct AI integration

Clear Entry Points

Discovery is streamlined via the “My Tickets” area and post-purchase CRM emails.

Users can easily see available actions for each ticket and identify which tickets are eligible for upgrades through an intuitive, dedicated interface.

A/B test showed outstanding results, with ~92% of users successfully discovering and understanding this feature.



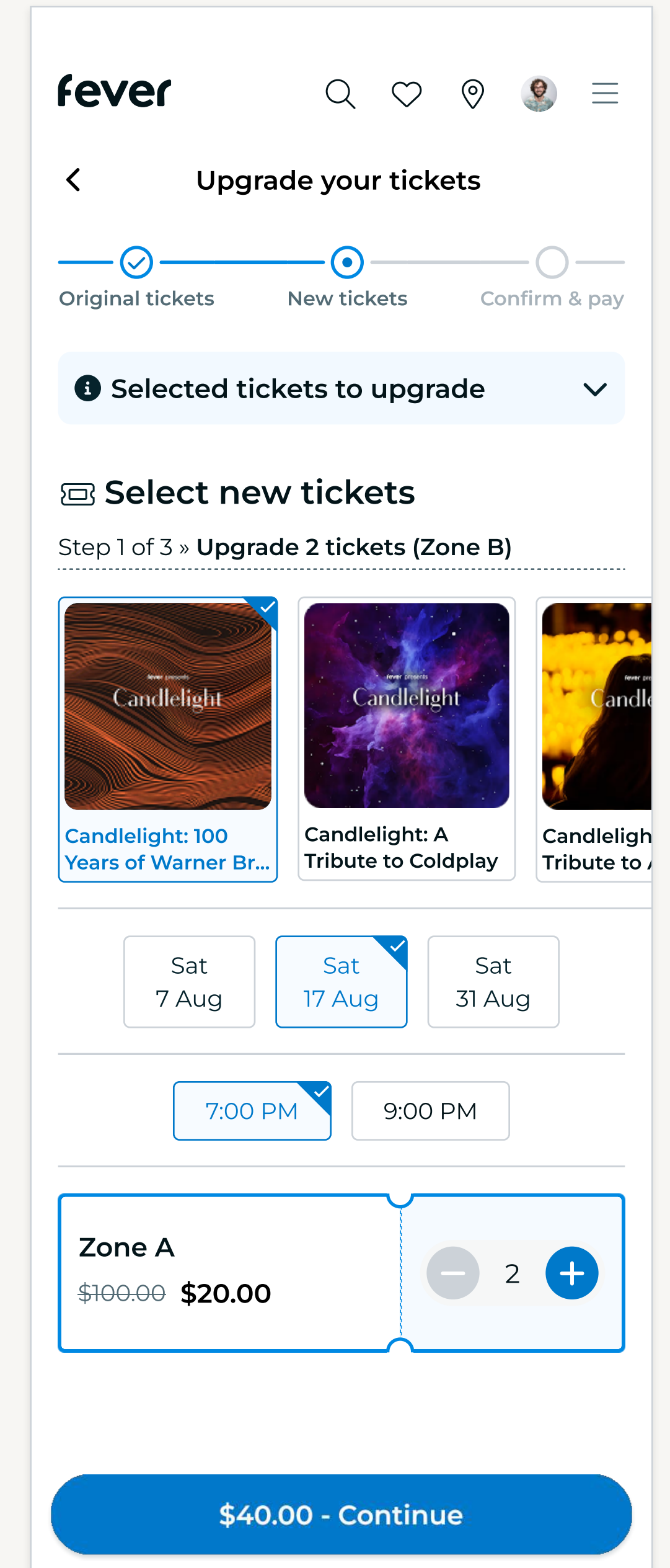
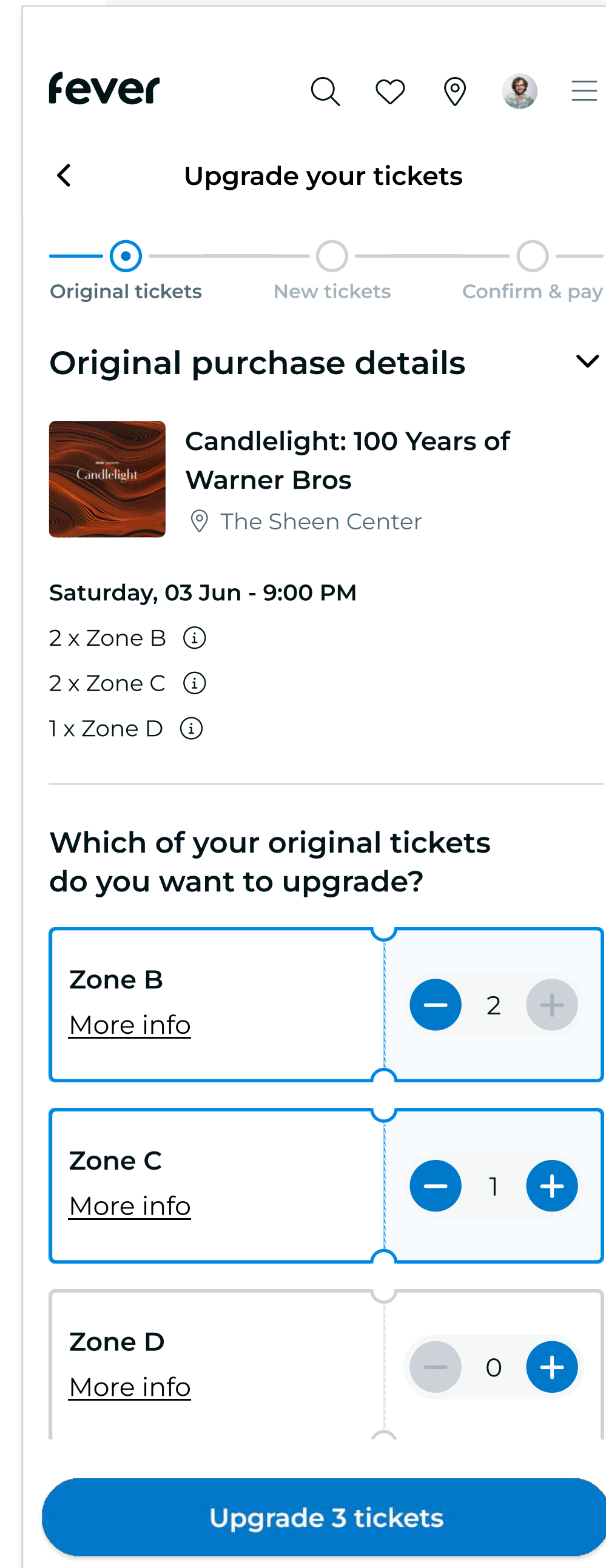
Tier Selection

A clear, guided choice architecture helps users easily select which tickets to upgrade, reducing friction and simplifying decisions.

Users pick the exact tickets and see only the available upgrade options.

They can also change associated plan IDs, dates, and hours within the same flow, avoiding the need for separate processes.

Downgrades are disabled to prevent leftover balances and ensure a smooth experience.

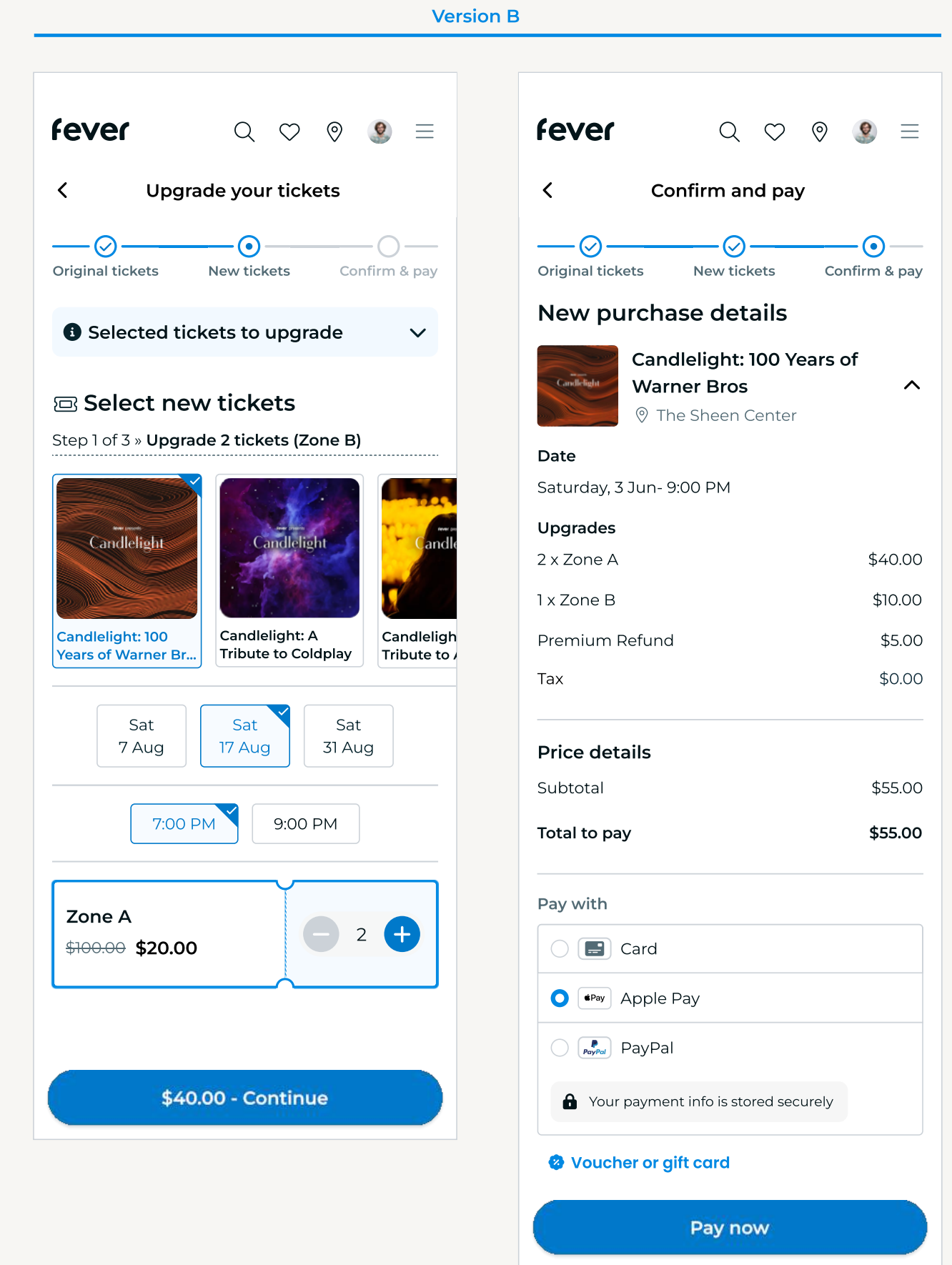
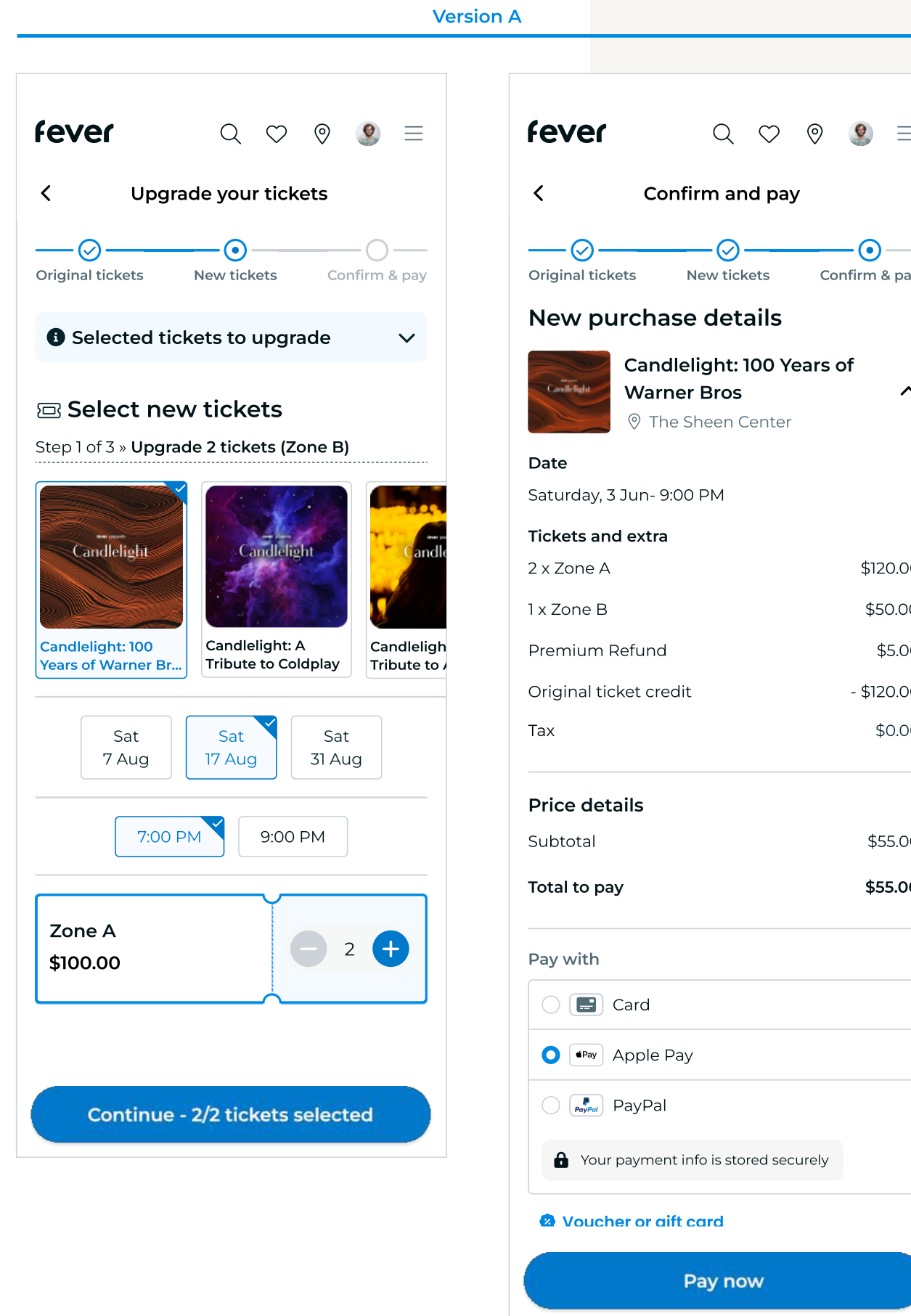


A/B tests on price framing

We ran controlled experiments keeping layout, steps, and eligibility constant, varying only how pricing was presented to identify the most transparent and high-converting approach.

Version A – Displayed full ticket prices throughout, with the original purchase applied as credit and the final amount clarified at checkout.

Version B – Displayed only the price difference (delta) at each step, keeping the additional cost immediately clear.



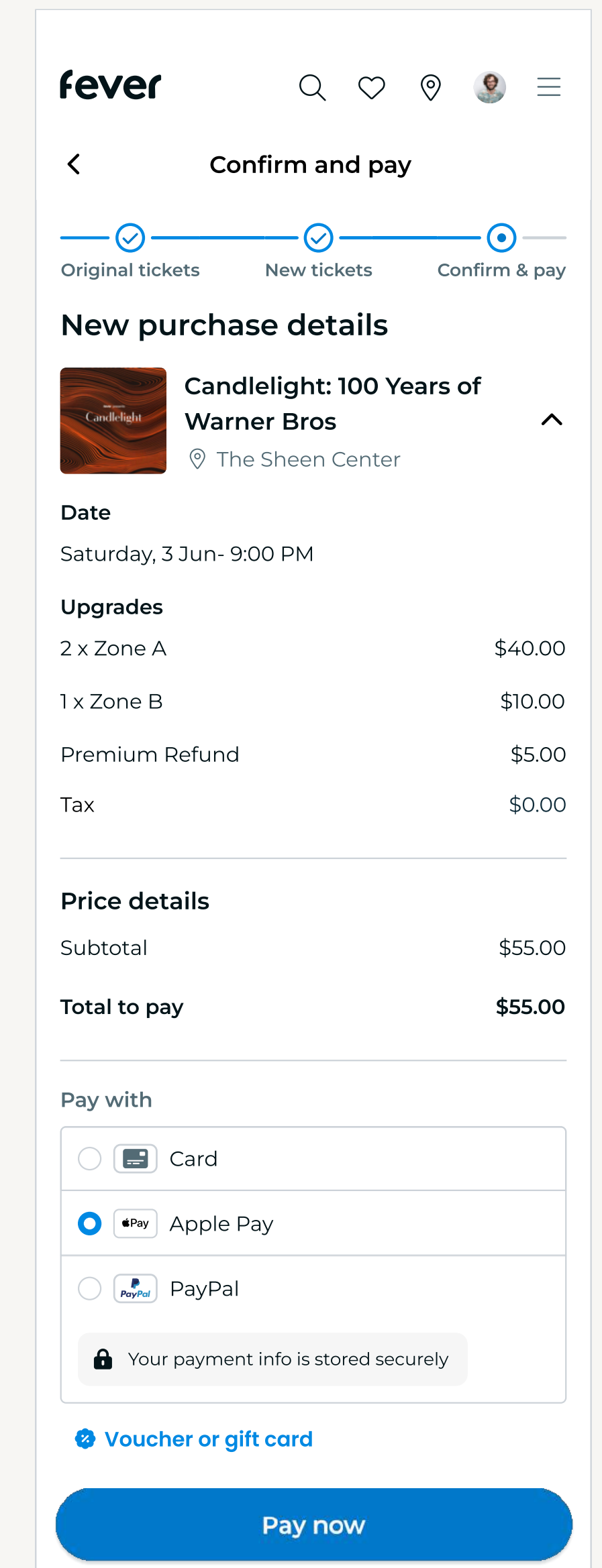
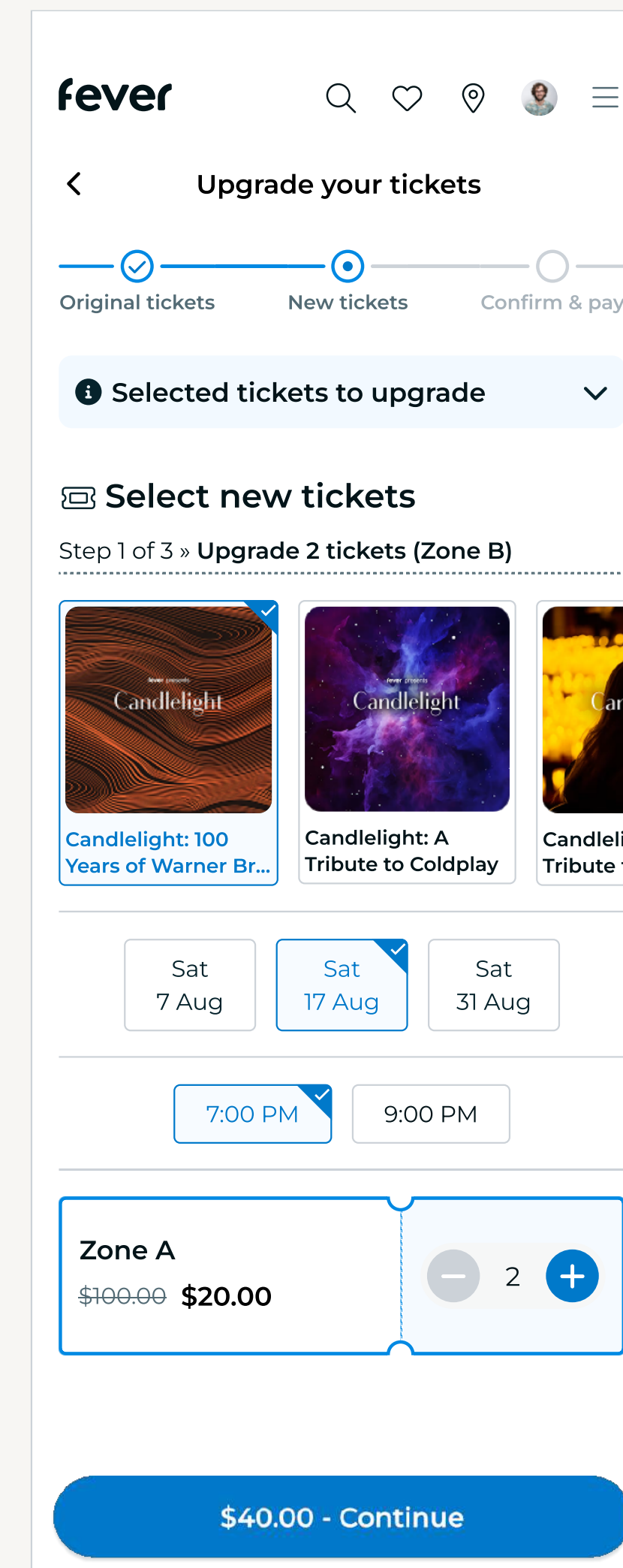
Price framing results

The delta-only funnel (**Variant B**) beat full list price with settlement at checkout (Variant A): **conversion was ~68% higher and abandonment was ~32% lower** from first eligible screen through payment. The lift held when we sliced by device and by upgrade size (small vs large delta).

SIGNAL	VARIANT B VS A (ROUNDED)
Funnel conversion (eligible → paid upgrade)	+68% relative lift
Abandonment (started flow, did not pay)	-32%
Median time to complete checkout	-14% (faster)
Average incremental revenue per completing session	+21%

50/50 split, ~18k eligible sessions over three weeks on matched plans and locales; uplift significant at 95% confidence on conversion and abandonment.

Version B



Seamless Delta Payment

Users pay only the price difference when upgrading, eliminating refunds or repurchases and creating a smoother transaction experience.

Users see the total amount they'll pay for the selected upgrades before confirming.

The payment page shows only the upgrade cost, while the original ticket price remains accessible in the "Ticket Detail" view.

Zone A
~~\$100.00~~ \$20.00

− 1 +

\$20.00 - Continue

Zone A
~~\$100.00~~ \$20.00

− 2 +

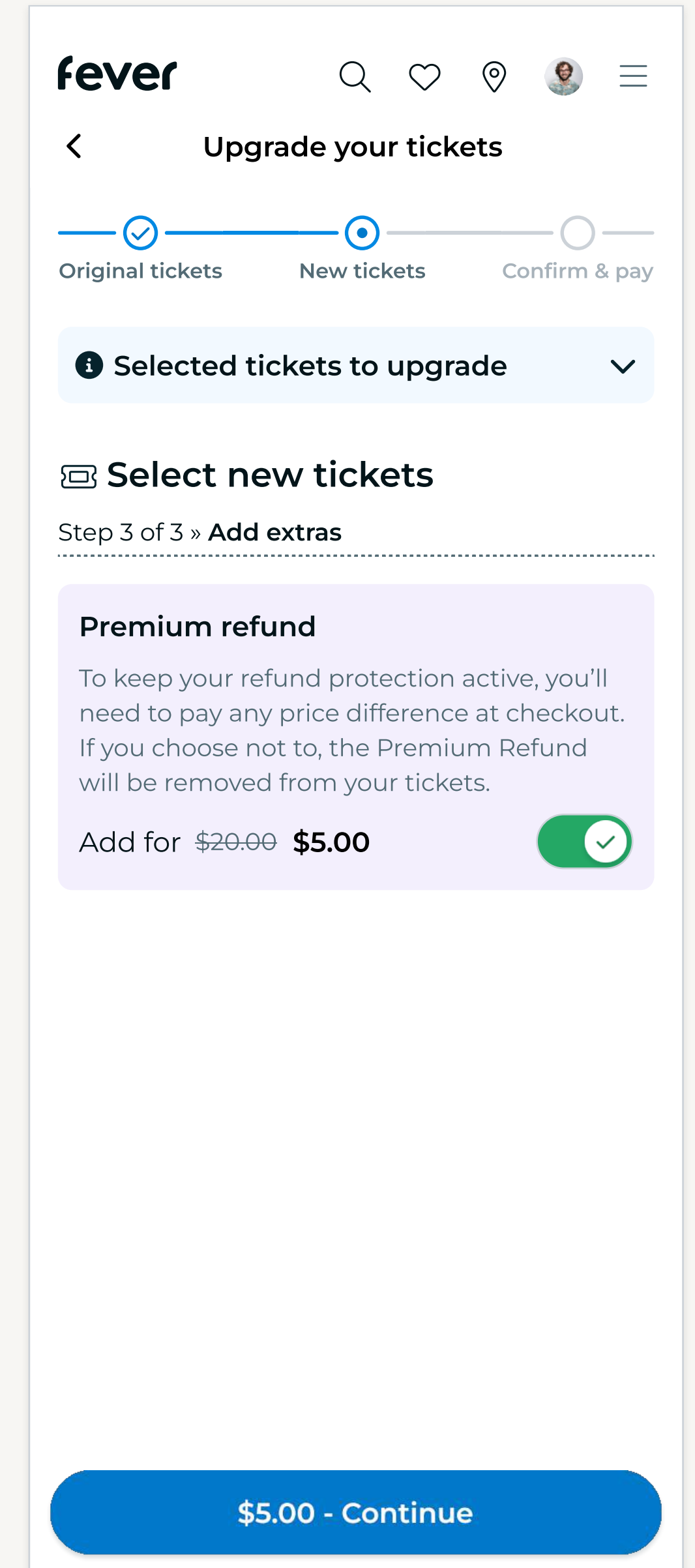
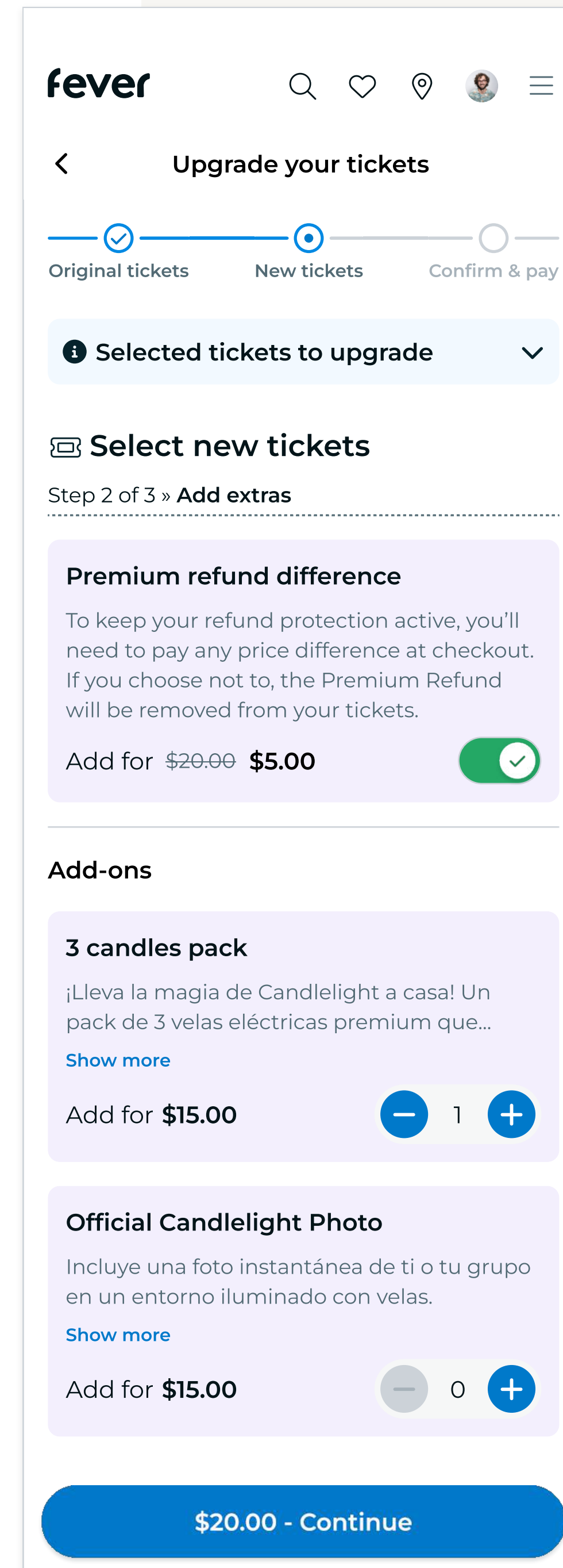
\$40.00 - Continue

The screenshot shows the 'Confirm and pay' screen in the fever app. At the top, there's a progress bar with three steps: 'Original tickets' (checked), 'New tickets' (checked), and 'Confirm & pay' (active). Below the progress bar, the event details are shown: 'Candlelight: 100 Years of Warner Bros' at 'The Sheen Center' on 'Saturday, 3 Jun - 9:00 PM'. A list of upgrades is displayed: '2 x Zone A' for \$40.00, '1 x Zone B' for \$10.00, 'Premium Refund' for \$5.00, and 'Tax' for \$0.00. The 'Price details' section shows a 'Subtotal' of \$55.00 and a 'Total to pay' of \$55.00. Under 'Pay with', there are three options: 'Card', 'Apple Pay' (selected), and 'PayPal'. A security notice states 'Your payment info is stored securely'. At the bottom, there is a 'Pay now' button.

Add-ons and extra

We introduced the ability for users to purchase add-ons and extras during upgrades, something previously unavailable.

This approach maximizes event conversion and revenue by offering upgrades, add-ons, and extras at advantageous prices.



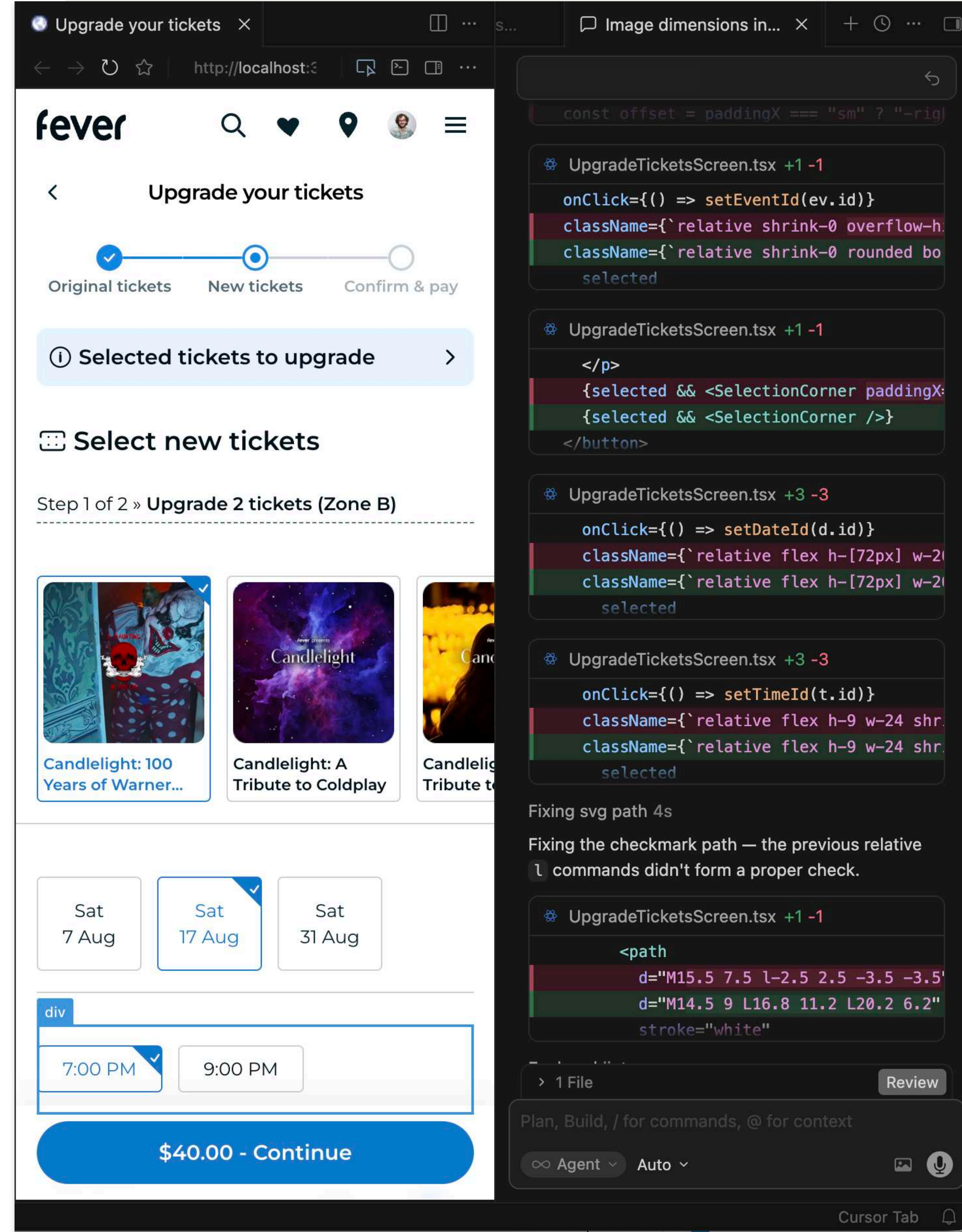
AI as workflow accelerator

We've used Cursor as our main tool across the entire process, from discovery to design and handoff.

We've fundamentally **sped up our workflow**. Instead of constantly looping between requirements, design, and technical constraints, we relied on a **single tool to inform and shape the entire system**. Design, product, and engineering stayed aligned through a shared, evolving prototype.

All flows, interactions, and edge cases are handled directly within Cursor, making them an **official part of the handoff** rather than something interpreted later, reduced developer confusion and sped up the launch.

Figma now plays a more focused role as a guide or blueprint. Only the core screens are fully designed to provide precise visual direction. (There are still occasional inconsistencies at the level of spacing and fine details that need an hardcoded solution.)

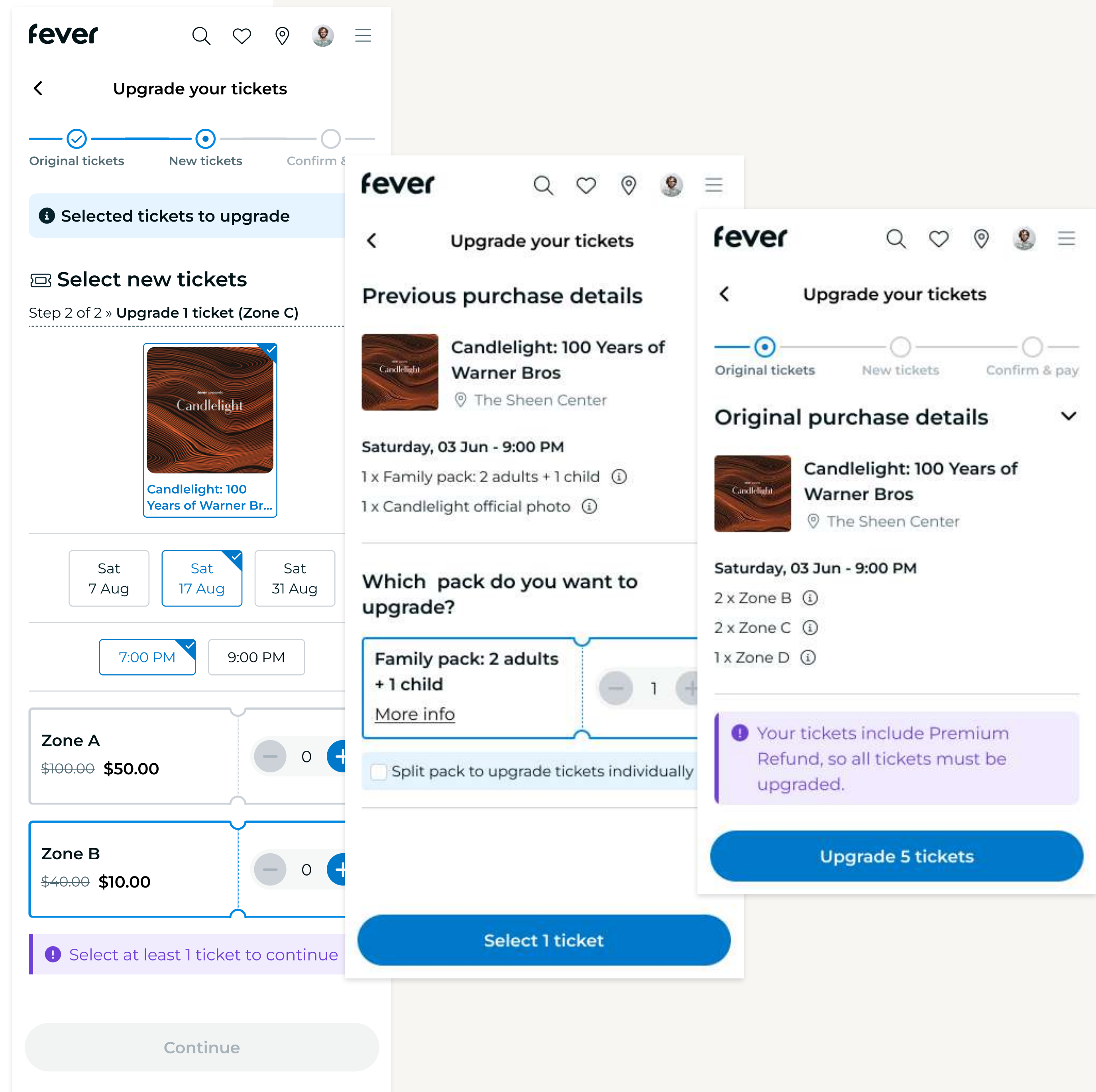


Edge Cases and Variants Managed with AI

Edge cases and flow variants are automatically handled through Cursor, reducing the need to design every screen or variant in Figma.

Examples include:

- Events with pre-paid extras (e.g., “Premium Refund”) require all tickets in the bundle to be upgraded together.
- Ticket packs can be upgraded as a whole or split to follow the regular upgrade flow.
- Users selecting fewer tickets than initially checked are guided correctly through the process.



create → use → validate → merge → update system

Tokens & Design System

We are also using Cursor to **continuously evolve our design system** alongside product development, ensuring that design and implementation stay fully aligned.

- A shared token library is defined and maintained in Cursor, acting as a source of truth for the entire team.
- All teams reference this library when building new features.
- When a project is completed, its output is pushed to the main repository and goes through a validation step before integration.

This serves two purposes:

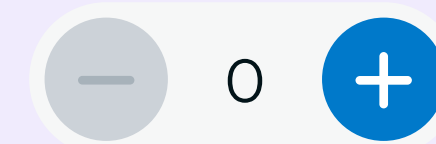
- **Coherence check**: ensuring that what has been produced is consistent with the overall product architecture, design patterns, and existing components
- **Design System update**: identifying any new components or patterns introduced during the project

Official Candlelight Photo

Incluye una foto instantánea de ti o tu grupo en un entorno iluminado con velas.

[Show more](#)

Add for **\$15.00**



Premium refund

To keep your refund protection active, you'll need to pay any price difference at checkout. If you choose not to, the Premium Refund will be removed from your tickets.

Add for ~~\$20.00~~ **\$5.00**



A continuous feedback loop:

- Features are built using shared tokens and components
- Outputs are validated against the full system
- The Design System is updated in a controlled way
- Updates are instantly available across all teams

The implemented solution delivered strong results, generating overwhelmingly positive user feedback and significantly increasing upgrade rates.

+3M €

Annual incremental revenue

+68%

Completion rate

-44%

Median time to complete checkout

2%

Support baseline reduced to

Future outlook

Upgrades and rescheduling for plans with real seating maps, and finer-grained PRP so people can split or tweak protection per line during an upgrade.

Phase 2, GA GA

Global availability across all eligible Marketplace plans, once the pilot clears the bar.

METRIC	BASELINE (MVP)	TARGET
Upgrade adoption (user)	0.27%	1.0%
Gross revenue increase	0.45%	> 0.5%
Feature conversion rate	13%	20%
Support ticket volume (exchanges)	Prior baseline	~2% of prior baseline
Plan coverage	20 plans	Expanding share of eligible plans globally

Key learnings and takeaways

Cognitive Load Over Full Disclosure

Transparency doesn't always mean showing *everything* at once. While Variant A showed the full list price and settlement credit, it forced users to perform mental math. Variant B's **68% relative lift** in conversion proved that "Delta-only" pricing aligns better with user psychology by focusing solely on the immediate incremental cost.

Design as an Operational Lever

This project highlighted that product design is a powerful tool for cost reduction. By moving upgrades from a manual support task to a product-led path, we didn't just capture revenue; we targeted a reduction in support ticket volume to a 2% baseline. Success is measured by the load taken off the User Operations team as much as the \$3M+ in incremental revenue.

Strategic Risk Mitigation through AI

Leveraging AI for real-time requirement definition allowed us to bypass traditional bottlenecks. By using AI to analyze and summarize support logs and user data, we built a live prototype that informed the PRD iteratively, rather than waiting for a static document.

New real time collaboration (AI as framework accelerator)

We accelerated our workflow by leveraging AI as a framework, drastically reducing bottlenecks and improving communication across teams. This allowed designers, developers, and product managers to collaborate in real time, make faster decisions, and deliver features more efficiently without waiting for sequential handoffs.